Draw it or Lose it

CS 230 Project Software Design Template

Version 1.0

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	9/10/2023	Joseph Marek	Added information for the initial design project

Instructions

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

Executive Summary

Our new client, The Gaming Room, is eager to create a web-based game that can be accessible across various platforms. They have their eyes set on "Draw It or Lose It," which currently operates exclusively on Android. The company's ambition is to make this game compatible with different operating systems, including MacOS, Linux, Windows, and more. The game involves teams competing to identify the images being drawn, with time constraints for both guessing and drawing. Each game session comprises four rounds, each lasting one minute.

Requirements

Design Constraints

1. Cross-Platform Compatibility: Currently an Android app, the goal is to make the game accessible on various platforms. To achieve this, we'll transform the app into a web application, ensuring compatibility with different operating systems. This can be achieved by implementing a REST API to facilitate communication via HTTP, thereby avoiding the need to choose a platform-specific language.

2. User Interface Consistency: As the game is already deployed as an Android application, maintaining a similar, if not identical, interface design for users is essential. Alternatively, there's an option to enhance the design and present it as a brand-new game.

3. Support for Multiple Teams and Players: The game must accommodate more than one team, each with multiple players. To achieve this, a client-server architecture should be employed, ensuring that the server can efficiently handle multiple players concurrently.

4. Unique Identifiers: Games and team names need to be unique, with only one game existing in memory at any given time. To address this, the incorporation of unique identifiers (IDs) for games, teams, and players is imperative. Additionally, a strategy for efficient memory allocation needs to be devised, considering the application's availability on multiple platforms. Platform-specific player IDs should also be managed.

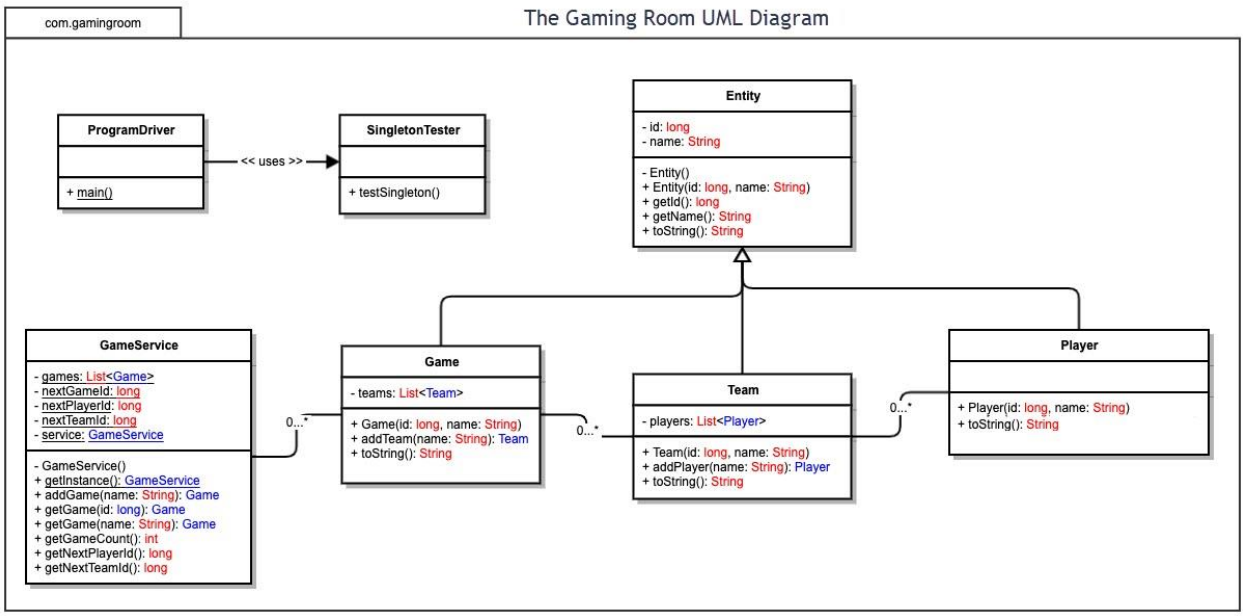
5. Images and Copyrights: The Gaming Room intends to utilize images in the Draw It or Lose It game. Existing images from the Android platform can be adopted and made compatible with all other platforms. However, it's crucial to secure licenses or copyright permissions for any additional images incorporated during the design process to ensure legal compliance.

System Architecture View

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

Domain Model

We will utilize the Unified Modeling Language (UML) diagram to provide a visual representation of the game system's design. Let's take a closer look at how the program will be developed: In the UML diagram, the entity class establishes relationships between the game, team, and player classes. The arrows signify that all these classes will inherit attributes from a superclass. This diagram presents the classes, variables, and methods that will be employed during development. The programDriver Class, located in the top left corner, is connected to the singletonTester. This indicates that the programDriver will employ the SingletonTester to test the code. This testing mechanism is implemented to ensure compliance with the requirement of having only one instance of the game existing in memory. The GameService class will encompass all the intricate methods that constitute the core of the game and its functionality. Meeting the program's requirement for unique games, teams, and players necessitated the design of dedicated classes, as depicted in this diagram. The lines connecting each class signify their associations, and the numbers between the lines denote the quantity of associations within each class. For instance, the GameService can have zero or more games associated with the Game class, and similar associations apply to the other classes. This comprehensive diagram will serve as a guiding framework during the development process, allowing for the creation of the final product. Adjustments and refinements may be made as necessary along the way to ensure the successful realization of the game system.



Evaluation

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client's requirements and look at the situation holistically, as it all has to work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

Development Requirements	Mac	Linux	Windows	Mobile Devices

Server Side	Apple, the creator of MacOS, provides server-based solutions that can be highly beneficial for Mac users within a network. These benefits include robust support for Mac applications and user-friendly administration through an excellent graphical interface. However, it's important to note that using a Mac server may have downsides, such as higher maintenance costs and limited suitability for large corporations or enterprises that heavily rely on third-party software and customization.	Linux provides numerous benefits for web hosting, with even major companies like Google opting for Linux servers. The primary advantage of using a Linux server is its cost-effectiveness, as it is both free and open source. This means that deploying and using tools can be inexpensive or even free. Another key benefit is its customizable security, allowing organizations to tailor security measures to their specific needs. Additionally, Linux web hosting supports popular programming languages such as Python, PHP, Perl, and Ruby. However, drawbacks of using a Linux server include a potentially steep learning curve for those without prior experience, potential compatibility issues with certain applications, and challenges when migrating from Windows to Linux.	Windows is a comprehensive operating system, but it comes with licensing costs as it's proprietary software. Despite the potential expense, there are notable benefits to using a Windows web hosting server. One advantage is its extensive support for various applications and third-party software. Additionally, Windows offers straightforward patch and hardware updates. Lastly, for developers familiar with the platform, it provides full support for script languages like ASP.NET and databases like MySQL.	While mobile devices for web servers are not widely recognized and may be seen as impractical, it is indeed a feasible concept. Companies like Oracle provide solutions for mobile server-side implementation. Oracle Database Mobile Server, for instance, can efficiently oversee applications, users, devices, and data across extensive deployments of mobile or remote devices. The advantages of adopting Oracle's database structure include robust support for iOS and Android development tools, as well as seamless synchronization capabilities with NoSQL Oracle databases.

Client Side	Developing for the Mac platform offers the advantage of user-friendly and robust software development kits (SDKs). However, a notable requirement is the necessity of owning a Mac Book for Mac development. This can result in added expenses and demands a developer who is proficient in Swift, the primary programming language for macOS and iOS applications.	The primary cost associated with this approach is the investment in development time. Additionally, it's crucial to have a team member experienced in using Python to effectively implement this solution.	Expertise is indeed a critical requirement when working with Windows. It is highly advisable to leverage the .NET framework for enhanced security and expanded capabilities when developing Windows-based solutions.	When it comes to mobile app development, it's essential to collaborate with developers who possess experience in building mobile applications. Developing for mobile devices requires a distinct approach as user interaction and the presentation of content differ significantly from web-based applications. Experienced mobile app developers are well-versed in crafting user-friendly and visually appealing interfaces tailored to the unique characteristics of mobile devices.

Development Tools	To develop using Swift on a Mac Book, you'll need a Mac Book running macOS, and you can use Xcode as your integrated development environment (IDE) for coding. Swift is the primary programming language for developing applications on macOS and iOS, and Xcode is Apple's official IDE for Swift development. With these tools, you can create a wide range of applications for the Apple ecosystem, including macOS, iOS, watchOS, and tvOS apps.	Python is commonly pre-installed on many Linux distributions, making it easily accessible for development. If you prefer using the IntelliJ IDEA Ultimate IDE for coding in Python on a Linux system, you can certainly do so. IntelliJ IDEA is a popular integrated development environment with a dedicated Python plugin that provides robust features for Python development, making it a valuable tool for coding and project management in the Python programming language.	Visual Studio Code (VS Code) is indeed a widely used and highly regarded code editor for Windows application development. While it's a versatile code editor that supports a variety of programming languages, C++ and C# are two of the most commonly used languages for developing Windows applications. Microsoft's Visual Studio IDE also offers comprehensive tools for Windows application development, particularly for C# developers working on .NET-based applications. The choice of language depends on the specific requirements and preferences of the project, but C++ and C# are certainly prominent choices for Windows development.	When it comes to mobile app development, there are several paths to consider. For Android app development, you'll want a specialist who is well-versed in Android Studio. To create iPhone apps, you'll need the expertise of someone with access to a Mac, capable of using Swift and Xcode. Alternatively, you can opt for Unity, which utilizes C++. This choice allows you to develop an app that can be converted for use on both Android and iPhone devices. However, it's important to note that to convert the app for iPhone, access to a Mac is still required to complete the final steps in Xcode.

Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

1. **Operating Platform**: I strongly recommend utilizing Linux Ubuntu Server as your hosting platform for deploying Draw It or Lose It on a Kubernetes cloud setup.

2. **Operating Systems Architectures**: The Linux kernel boasts both stability and security, making it a reliable choice for hosting purposes. Additionally, Kubernetes clusters provide a convenient means to segregate system and hardware requirements, further enhancing the overall robustness and flexibility of your hosting environment.

3. **Storage Management**: You have the option to choose between HDD (Hard Disk Drive) storage or SSD (Solid State Drive) storage, but I highly recommend the latter, SSD storage. SSDs offer faster access to assets, which is crucial for efficiently loading pictures onto users' devices, thereby enhancing the overall user experience. To optimize your storage setup, I recommend establishing a Kubernetes node specifically for file storage and a NoSQL node dedicated to game data and user management. In the NoSQL database, you can link the URLs to the location of the pictures, ensuring efficient retrieval and management of assets.

4. **Memory Management**: To help manage costs effectively, I suggest implementing a system load watcher. This way, you can dynamically adjust the allocated system resources based on usage patterns. During periods of low usage, you can reduce the required memory, and during peak times, you can scale up as needed to ensure the best user experience. This approach allows you to optimize your expenses by only paying for the resources that are necessary at any given time.

5. **Distributed Systems and Networks**: Having your system in the cloud offers significant advantages, particularly in terms of scalability and fault tolerance. With cloud-based infrastructure, you can easily handle server maintenance, scaling, and even server failures without interrupting the game. If there's a need for resource minimization, you can migrate nodes to other servers, or in case of a server crash, automated failover mechanisms can ensure seamless continuity. By adopting this cloud-based approach, you'll be able to host all the necessary components to run the game, leaving the client end with the sole responsibility of accessing this information. This setup ensures compatibility with various operating systems, allowing you to create clients that can access the game's data. Leveraging Kubernetes for function separation offers streamlined management and organization of your system, contributing to a more efficient and resilient infrastructure.

6. **Security**: I highly recommend implementing a role-based security system for your application. This approach provides a flexible and effective way to manage access and separation of roles, including admin, game, team, player, and user. By using role-based security, you can ensure that users can only access the information and functionalities that are relevant to their specific roles, enhancing data security and user privacy.